**Table of Contents**

1. **Introduction**
2. **Overview**
3. **Specifications**
4. **System Architecture and Design**
5. **Connector pin-outs and cables**
6. **Software design**
7. **Introduction**

This document is the System Reference Manual for the WINGZ Board, a low cost OMAP3530 based board. This document provides detailed information on the overall design and usage of the WINGZ Board from the system level perspective. It is not intended to provide detailed documentation of the OMAP3530 processor or any other component used on the board. It is expected that the user will refer to the appropriate documents for these devices to access detailed information.

1. **Overview**

Recent times ZigBee based wireless sensor networks are gaining popularity in various application domain areas like healthcare, structural health monitoring, home and commercial building automation, industrial automation, transportation, agriculture etc. ZigBee standard provides various domain specific application profiles. The sensors are interfaced to ZigBee communication modules to form a wireless personal area network. In order to control and communicate to these networks from backbone infrastructure networks viz. 2G/3G/4G/Wi-Fi a wireless enabled gateway is essential. The hardware platforms and software framework for such Gateways need to be application domain independent to make the systems generic. The work that has been presented in this paper focuses on hardware system design issues and options, software framework and application case studies. The gateway subsystem has been enabled with sufficient storage for local data buffering. The system consists of three modules namely WINGZ (Wireless IP Network Gateway for Zigbee), Ubimote (Wireless ZigBee mote with generic sensor interface) and Ubi-Sense (Generic Sensor board) for application case study. The Gateway Related hardware and software systems enable anywhere connectivity of networked sensors. The heart of the system is WINGZ which is an android based multi protocol wireless gateway supporting WLAN, HSPA (High Speed Packet Access), ZigBee and Ethernet.

1. **Specifications**

This section covers the specifications of the WINGZ Board and provides a high level description of the major components and interfaces that make up the WINGZ Board.

* 1. OMAP Processor

The WINGZ Board uses the OMAP3530DCBB72 720MHZ version and comes in a .4mm pitch POP package. POP (Package on Package) is a technique where the memory, NAND and SDRAM, are mounted on top of the OMAP3530. For this reason, when looking at the WINGZ Board, you will not find an actual part labeled OMAP3530.

* 1. Memory

The Micron POP memory is used on the WINGZ Board and is mounted on top of the processor as mentioned. The key function of the POP memory is to provide:

* + 2Gb NAND x 16 (256MB)
  + 2Gb MDDR SDRAM x32 (256MB @ 166MHz)

No other memory devices are on the WINGZ Board. It is possible however, that additional memory can be added to WINGZ Board by:

* + Installing a SD or MMC in the SD/MMC slot
  + Use the USB OTG port and a powered USB hub to drive a USB Thumb drive or hard drive.
  + Install a thumbdrive into the EHCI USB port Support for this is dependent upon driver support in the OS.

1. **System Architecture and Design**
2. **Connector pin-outs and cables**
3. **Software design**
   1. **Installing packages**

Install these packages before proceeding further

*sudo apt-get update*

*sudo apt-get install libc6:i386 libstdc++6:i386 libncurses5:i386 zlib1g:i386*

Do the following steps in the followiong folder

**~/**

git clone <https://github.com/CDACBANG/WINGZ-FILES.git>

**Get and Extract the compiler**

*wget -c* [*https://releases.linaro.org/14.09/components/toolchain/binaries/gcc-linaro-arm-linux-gnueabihf-4.9-2014.09\_linux.tar.xz*](https://releases.linaro.org/14.09/components/toolchain/binaries/gcc-linaro-arm-linux-gnueabihf-4.9-2014.09_linux.tar.xz)

*tar xf gcc-linaro-arm-linux-gnueabihf-4.9-2014.09\_linux.tar.xz*

*export CC=`pwd`/gcc-linaro-arm-linux-gnueabihf-4.9-2014.09\_linux/bin/arm-linux-gnueabihf-*

**Verify the proper version:**

*${CC}gcc --version*

arm-linux-gnueabihf-gcc (crosstool-NG linaro-1.13.1-4.9-2014.09 - Linaro GCC 4.9-2014.09) 4.9.2 20140904 (prerelease)

Copyright © 2014 Free Software Foundation, Inc.

This is free software; see the source **for** copying conditions. There is NO

warranty; not even **for** MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

6.2. **U-Boot Compilation for WINGZ board**

For u-boot:

*cd* WINGZ-FILES/wingz\_uboot/uboot/

**../u-boot/**

*make ARCH=arm CROSS\_COMPILE=${CC}* distclean

*make ARCH=arm CROSS\_COMPILE=${CC}* omap3\_wingz\_defconfig

*make ARCH=arm CROSS\_COMPILE=${CC}*

**6.3 KERNEL Compilation for WINGZ Board**

**~/**

Get the latest kernel:

*git clone https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git*

copy **omap3-wingz.dts** to the dts folder of arm

*cp* ../wingz\_dts/omap3-wingz.dts ~/linux-stable/arch/arm/boot/dts/

added entry in linux-stable/arch/arm/boot/dts/Makefile:

-

-

-

dtb-$(CONFIG\_ARCH\_OMAP3) += am3517-craneboard.dtb \

am3517-evm.dtb \

am3517\_mt\_ventoux.dtb \

omap3430-sdp.dtb \

omap3-beagle.dtb \

**omap3-wingz.dtb \**

omap3-beagle-xm.dtb \

omap3-beagle-xm-ab.dtb \

omap3-cm-t3517.dtb \

omap3-cm-t3530.dtb \

-

-

-

*make ARCH=arm CROSS\_COMPILE=${CC}* *distclean*

*cp ../*wingz\_config/wingz-config ~/linux-stable/.config

*make ARCH=arm CROSS\_COMPILE=${CC}* oldconfig

*make ARCH=arm CROSS\_COMPILE=${CC}* zImagemodules

*make ARCH=arm CROSS\_COMPILE=${CC}* dtbs

**zImage** will be created in ~/linux-stable/arch/arm/boot/ folder

**After compilation:-**

This will export the kernel\_version you just compiled

export kernel\_version=$(cat include/generated/utsrelease.h | awk '{print $3}' | sed 's/\"//g' )

*cp* -v arch/arm/boot/zImage "~/compiled\_modules/kernel\_image/${kernel\_version}.zImage"

*cp* -v .config "~/compiled\_modules/kernel\_image/config-${kernel\_version}"

*make -s ARCH=arm CROSS\_COMPILE="${CC}"* modules\_install INSTALL\_MOD\_PATH=~/compiled\_modules/modules/

*make -s ARCH=arm CROSS\_COMPILE="${CC}"* firmware\_install INSTALL\_FW\_PATH=/~/compiled\_modules/firmwares/

**copy bootlaoder and image to SD card:**

*sudo cp -v* ./u-boot-wingz/MLO /media/boot/

*sudo cp -v* ./u-boot-wingz/u-boot.img /media/boot/

**File system:**

Get:

wget -c <https://rcn-ee.net/deb/minfs/trusty/ubuntu-14.04.1-minimal-armhf-2014-11-10.tar.xz>

Verify:

md5sum ubuntu-14.04.1-minimal-armhf-2014-11-10.tar.xz

7e5fa3cb4814f195d75cbaad6d922090 ubuntu-14.04.1-minimal-armhf-2014-11-10.tar.xz

Extract:

tar xf ubuntu-14.04.1-minimal-armhf-2014-11-10.tar.xz

**or**

use fs in ../WINGZ-FILES/wingz\_fs/ubuntu-14.04.1-minimal-armhf-2014-11-10.tar.xz

tar xf ubuntu-14.04.1-minimal-armhf-2014-11-10.tar.xz

**FS to SD card:**

*sudo tar xfvp* ../ubuntu-14.04.1-minimal-armhf-2014-11-10/armhf-rootfs-\*.tar -C /media/rootfs/

sudo mkdir -p /media/rootfs/boot/

sudo sh -c "echo 'uname\_r=${kernel\_version}' > /media/rootfs/boot/uEnv.txt"

**Install kernel to SD card:**

sudo cp -v ~/compiled\_modules/kernel\_image/${kernel\_version}.zImage /media/rootfs/boot/vmlinuz-${kernel\_version}

sudo mkdir -p /media/rootfs/boot/dtbs/${kernel\_version}/

sudo cp -rfv ~/compiled\_modules/dtbs/ /media/rootfs/boot/dtbs/${kernel\_version}/

sudo cp -rfv ~/compiled\_modules/modules/\* /media/rootfs/

sudo cp -rfv ~/compiled\_modules/firmwares/\* /media/rootfs/lib/firmware/

**File system table:**

sudo sh -c "echo '/dev/mmcblk0p2 / auto errors=remount-ro 0 1' >> /media/rootfs/etc/fstab"

**Networking:**

sudo nano /media/rootfs/etc/network/interfaces

Add:

**/etc/networking/interfaces**

auto lo

iface lo inet

loopback

auto eth0

iface eth0 inet dhcp

hwaddress ether xx:xx:xx:xx:xx:xx

**Note**: use you own mac id for eth0 in order to avoid getting fake mac for every reboot

Note: After connecting LAN cable to the board if you are not getting ip run,

sudo /etc/init.d/networking restart

**Serial Login**

sudo nano /media/rootfs/etc/init/serial.conf

Add:

**/etc/init/serial.conf**

start on stopped rc RUNLEVEL=[2345]

stop on runlevel [!2345]

respawn

exec /sbin/getty 115200 ttyO2

sync

umount /media/boot

umount /media/rootfs